

WHAT IS CLAIMED IS:

1. A method for searching for programming objects that are available in a computing environment, the method comprising:

receiving instructions to search for objects, the instructions including at least one optional attribute that the objects should but are not required to have;

locating objects in the computing environment; and

ordering the located objects in a list of objects based on matches between the attributes of the located objects and at least one optional attribute.

2. The method of claim 1 wherein receiving instructions to search for objects comprises receiving at least one required attribute with the instructions, and wherein locating objects comprises locating objects that have the at least one required attribute.

3. The method of claim 1 wherein receiving instructions to search for objects comprises receiving instructions to search for objects within a particular category of objects.

09874550-053404

4. The method of claim 1 wherein locating objects comprises searching for sets of object data beneath a registry key.

5. The method of claim 4 wherein searching for sets of object data beneath a registry key comprises searching beneath a registry key for a category.

6. The method of claim 5 wherein locating objects further comprises:

comparing a required attribute that was received with the instructions to an attribute in each set of object data; and

for each attribute that matches the required attribute, adding a reference to an object associated with attribute to the list of references.

7. The method of claim 1 wherein ordering references comprises ordering pointers to object tokens associated with the object.

8. The method of claim 7 wherein the object token points to a set of object data associated with the object.

9. The method of claim 8 wherein the set of object data includes a class identifier for the object.

09871550 053101

10. The method of claim 1 wherein locating objects comprises locating objects of an object class that is dynamically made available.

11. The method of claim 1 wherein locating objects comprises locating object data that is stored on a remote computer that is remote from a local computer on which the instruction to search was received.

12. The method of claim 11 wherein locating objects further comprises:

comparing a required attribute that was received with the instructions to an attribute on the remote computer; and
for each attribute that matches the required attribute, adding a reference to an object associated with the attribute to the list of references.

13. The method of claim 12 wherein comparing the required attributes to the attributes on the remote computer comprises:

instantiating an object token;
initializing the object token to point to the attributes on the remote computer;
calling a method on the object token to retrieve the attributes; and

09871550.053101

comparing the retrieved attributes to the required attributes.

14. The method of claim 12 wherein the remote computer and the local computer are on the same local area network.

15. The method of claim 12 wherein the remote computer and the local computer are on different local area networks.

16. The method of claim 12 wherein the remote computer and the local computer are connected together through the Internet.

17. A computer-readable medium having computer-executable instructions for performing steps comprising:

searching for sets of object attributes;
comparing found object attributes to a required attribute;

for those sets of object attributes that have an attribute that matches the required attribute, comparing the set of object attributes to an optional attribute; and

placing a reference to those object attributes that match the required attribute in a list such that the list is ordered based on the comparison

0987450 053104

between the object attributes and the optional attribute.

18. The computer-readable medium of claim 17 wherein comparing found object attributes to a required attribute comprises comparing found object attributes to a required attribute of null such that all found attributes match the required attribute.

19. The computer-readable medium of claim 17 wherein searching for sets of object attributes comprises searching for attributes in a registry in a local computer.

20. The computer-readable medium of claim 19 wherein searching for attributes in a registry comprises searching under a key assigned to a category of objects.

21. The computer-readable medium of claim 17 wherein searching for sets of object attributes comprises searching for sets of object attributes associated with object classes that are dynamically made available.

22. The computer-readable medium of claim 17 wherein searching for sets of object attributes comprises searching in a storage location on a remote computer.

0937150-053101

23. The computer-readable medium of claim 22 wherein searching in a storage location on a remote computer comprises searching through an Internet connection to the remote computer.

24. The computer-readable medium of claim 17 wherein comparing found object attributes to a required attribute comprises:

- instantiating an object token;
- initializing the object token to point to a set of found object data;
- calling a method on the object token to retrieve attributes from the set of found object data; and
- comparing the retrieved attributes to the required attribute.

25. The computer-readable medium of claim 17 wherein placing a reference in a list comprises:

- instantiating an object token;
- initializing the object token to point to a set of found object data; and
- placing a pointer to the object token in the list.

26. A method for searching for computer programming objects, the method comprising:

- receiving a request on a local computer to search for a programming object based on at least one search attribute; and

09871550-053101

based on the request, searching for object attributes associated with object classes that have an accessibility that is subject to change.

27. The method of claim 26 wherein receiving a request comprises receiving a request that indicates a category of programming objects for the programming object.

28. The method of claim 27 wherein searching for object attributes comprises utilizing a token enumerator designated for the category of programming objects.

29. The method of claim 28 wherein utilizing a token enumerator comprises instantiating the token enumerator.

30. The method of claim 29 wherein instantiating a token enumerator comprises causing the token enumerator to perform steps comprising:

locating object data associated with an object class that has an accessibility that is subject to change;

instantiating an object token for each set of located object attributes;

initializing each object token to point to a set of located object data; and

09874530-054104
FOI b5 b7C b6

returning a separate pointer to each instantiated object token.

31. The method of claim 26 further comprising:
comparing object attributes found during
the search for attributes to the at
least one search attribute; and
providing a reference to an object
associated with an object attribute if
the object attribute matches the
search attribute.

32. The method of claim 31 wherein comparing
object attributes comprises:

for each set of found object attributes,
instantiating an object token;
initializing each object token so that it
can locate a set of object data;
calling a method on each object token to
retrieve an object attribute from the
set of object data it has been
initialized to; and
comparing the retrieved object attribute to
the search attribute.

33. A computer-readable medium having computer-
executable instructions for performing steps
comprising:

09871550-053101

receiving an instruction to search for an object based on at least one search attribute;

determining that a set of object attributes could be located outside of a static attribute storage location;

searching for a set of attributes outside of the static attribute storage location; and

returning a reference to an object based on a set of attributes found outside of the static attribute storage location.

34. The computer-readable medium of claim 33 wherein the step of determining that a set of object attributes could be located outside of a static attribute storage location comprises finding a token enumerator name in a registry.

35. The computer-readable medium of claim 34 wherein the step of searching comprises:

instantiating a token enumerator based on information under the token enumerator name; and

causing the token enumerator to search for the set of attributes.

36. The computer-readable medium of claim 35 wherein the step of returning a reference comprises:

09871550-053101

instantiating an object token for a found
set of attributes;
initializing the object token to point to
the set of attributes; and
returning a pointer to the object token.

37. The computer-readable medium of claim 33
wherein the step of searching comprises searching
through an Internet connection.

38. The computer-readable medium of claim 33
wherein the step of receiving an instruction to
search comprises receiving a required search
attribute.

39. The computer-readable medium of claim 33
wherein the step of receiving an instruction to
search comprises receiving an optional search
attribute.

40. The computer-readable medium of claim 39
wherein the step of returning a reference comprises
returning a list of references that are ordered based
on matches between the optional search attribute and
attributes in found sets of attributes.

41. A method of instantiating and initializing
a programming object, the method comprising:
selecting an object data set for an object
from a plurality of object data sets

09374550-053104

for the object's class, each object data set including the same unique identifier for the object's class; instantiating the object based on the unique identifier; and initializing the object using at least one attribute in the selected object data set.

42. The method of claim 41 wherein selecting an object data set comprises:

locating the object data set; instantiating an object token; and initializing the object token to point to the object data set.

43. The method of claim 42 wherein instantiating the object comprises calling an instantiation method exposed by the object token.

44. The method of claim 43 wherein initializing the object comprises having the instantiation method of the object token call an initialization method exposed by the object to set a pointer to the object token in the object.

45. The method of claim 44 wherein initializing the object further comprises having the initialization method in the object access attributes through the pointer to the object token and having

09871550-053101

the initialization method use at least one accessed attribute to initialize the object.

46. A computer-readable medium having a computer-loadable data structure, the data structure comprising:

a first set of object data for an object class wherein the first set of object data comprises a first entry containing a unique identifier for the object class and at least one attribute of the object class; and

a second set of object data for the object class wherein the second set of object data comprises a second entry containing the same unique identifier for the object class as the first entry and at least one attribute of the object class that is different from the at least one attribute of the first set of object data.

47. The computer-readable medium of claim 46 wherein the unique identifier is a class identifier that can be used to instantiate the object.

48. The computer-readable medium of claim 46 wherein one of the attributes is a data file for initializing the object.

09871550-053101

49. The computer-readable medium of claim 46 wherein the first set of object data is located on a first computer and the second set of object data is located on a second computer.

50. The computer-readable medium of claim 46 wherein the first set of object data and the second set of object data are both found in a registry in a local computer.

51. The computer-readable medium of claim 50 wherein the first and second sets of object data are both found under a Tokens key.

52. A computer-readable medium having a computer-loadable object token comprising:

computer-executable instructions for
setting the object token to point to a
set of object data related to an
object;

computer-executable instructions for
retrieving attributes from the set of
object data; and

computer-executable instructions for
instantiating the object based on a
unique identifier in the set of
object data.

53. The computer-readable medium of claim 52 wherein the computer-loadable object token further

09871550-053101

54. The computer-readable medium of claim 53 wherein the computer-loadable object token further comprises computer-executable instructions to cause an instantiated object to initialize itself by passing requests to the object token to retrieve attributes from the set of object data.

```
computer-executable      instructions      for
    locating object attributes that are
    located outside of a static attribute
    storage location; and
```

computer-executable instructions for providing a reference to the object attributes.

```
computer-executable      instructions      for
    instantiating an object token;
```

```
computer-executable      instructions      for
initializing the object token to point
to the object attributes; and
```

computer-executable instructions to provide
a pointer to the instantiated object

-46-

token as the reference to the object
attributes.

09871550-053101